CmSc 250 Intro to Algorithms

Graph Algorithms - Topological Sort

1. Introduction

Topological sort: an ordering of the vertices in a directed acyclic graph, such that:

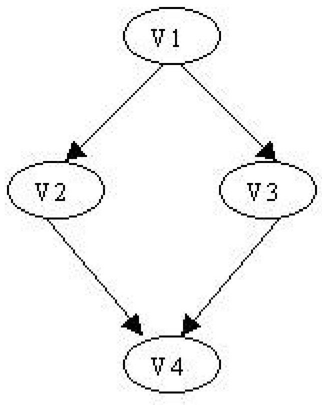
If there is a path from ***u*** to ***v***, then ***v*** appears after ***u*** in the ordering.

Types of graphs:

The graphs should be **directed:** otherwise for any edge (u,v)
there would be a path from ***u*** to ***v*** and also from ***v*** to ***u***,
and hence they cannot be ordered.

The graphs should be **acyclic**: otherwise for any two vertices ***u*** and ***v*** on a cycle
u would precede ***v*** and ***v*** would precede ***u.***

The ordering may not be unique:



V1, V2, V3, V4 and V1, V3, V2, V4 are legal orderings

Degree of a vertex U: the number of edges (U,V) - outgoing edges
Indegree of a vertex U: the number of edges (V,U) - incoming edges

The algorithm for topological sort uses "indegrees" of vertices.

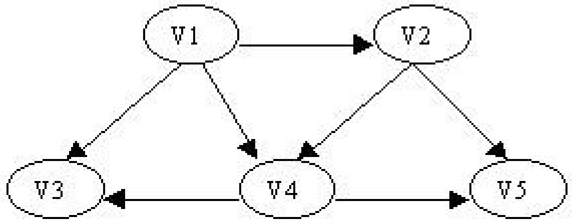
2. Algorithm

1. Compute the indegrees of all vertices
2. Find a vertex **U** with indegree 0 and print it (store it in the ordering)

   If there is no such vertex then there is a cycle
   and the vertices cannot be ordered. Stop.

3. Remove **U** and all its edges **(U,V)** from the graph.
4. Update the indegrees of the remaining vertices.
5. Repeat steps 2 through 4 while there are vertices to be processed.

3. Example



1. Compute the indegrees:

   V1: 0
   V2: 1
   V3: 2
   V4: 2
   V5: 2

2. Find a vertex with indegree 0: V1
3. Output V1 , remove V1 and update the indegrees:

   Sorted: V1
   Remove edges: (V1,V2) , (V1, V3) and (V1,V4)
   Updated indegrees:

      V2: 0
      V3: 1
      V4: 1
      V5: 2

The process is depicted in the following table:

	Indegree					
Sorted à		V1	V1,V2	V1,V2,V4	V1,V2,V4,V3	V1,V2,V4,V3,V5

V1	0					
V2	1	0				
V3	2	1	1	0		
V4	2	1	0			
V5	2	2	1	0	0	

One possible sorting: V1, V2, V4, V3, V5

Another sorting: V1, V2, V4, V5, V3

Complexity of this algorithm: O(|V|^2), |V| - the number of vertices.

To find a vertex of indegree 0 we scan all the vertices - |V| operations.

We do this for all vertices: |V|^2

4. Improved algorithm

After the initial scanning to find a vertex of degree 0, we need to scan only those vertices whose updated indegrees have become equal to zero.

1. Store all vertices with indegree 0 in a queue
2. **get** a vertex U and place it in the sorted sequence (array or another queue).
3. For all edges (U,V) update the indegree of V, and **put** V in the queue if the updated indegree is 0.
4. Perform steps 2 and 3 while the queue is not empty.

5. Complexity

The number of operations is O(|E| + |V|), where |V| - number of vertices, |E| - number of edges.
How many operations are needed to compute the indegrees?

Depends on the representation:
Adjacency lists: $O(|E|)$
Matrix: $O(|V|^2)$

Note that if the graph is complete $|E| = O(|V|^2)$

Learning Goals

- Be able to explain the basic idea of topological sort
- Be able to explain the improved version, using a queue and its complexity.
- Be able to run the improved algorithm on paper.

Exam-like questions

1. What is the complexity of topological sort using queues?
2. In what case the complexity of the improved version is $O(|V|^2)$?
3. What types of graphs can be subjected to topological sort?
4. Given a graph, run the algorithm on paper

Back to Contents page

Created by Lydia Sinapova